Second Banff Challenge, Problem 2
A nearest-neighbor approach

Doug Applegate[1*], Matt Bellis[1†],

[1]*Stanford University*

Abstract

We approach Problem 2 of the second Banff Challenge using a *nearest neighbor* algorithm to approximate the PDFs for the provided Monte Carlo samples. We bootstrap the MC samples to incorporate the uncertaties due to the finite samples and the correlations inherent in the estimation procedure. Toy studies are run to determine the susceptibility to false positives and the power to identify a true signal. The final results are then presented.

This study should not be taken as indicative of the standard statistical procedures used in the BaBar, LSST or any other collaboration with which the authors are affiliated.

*dapple@stanford.edu

†mbellis@stanford.edu

Contents

1 Statement of the problem

This study looks at the Second Banff Challenge, Problem 2, in which we are given 3 Monte Carlo samples of 5,000 events that represent two background samples and 1 signal sample. We use this information to develop a procedure to determine which, if any, of the 20,000 provided datasets contain a significant amount of signal.

We fit the datasets using a nearest neighbors algorithm to estimate the PDFs and a bootstrapping procedure to maginalize over the correlated uncertainties inherent in this approach. We use Monte Carlo studies (toy studies) to estimate the Type I error rate and the power of this procedure.

2 Theory and implementation

The second Banff challenge problem problem may be separated into three related inference problems. First, we must estimate the PDF functions for each of the three Monte Carlo channels provided. Second, we must infer the observed rates for each signal from the data. Last, we must report the presence, or lack thereof, of the signal channel.

2.1 Measuring PDFs from Monte Carlo

When determining the measured rates from each channel, we will need to calculate the probability that an event x_i came from a channel. We are not given the functional form of that PDF, but instead are presented with Monte Carlo samples drawn from the PDF. The standard response is to either choose an arbitrary functional form to fit to the Monte Carlo samples via an unbinned maximum likelihood approach or to histogram the samples to create a step function PDF. Both approaches introduce many new variables, *e.g.* choice of parametrization, or bin edges. Either approach may miss the pathological features seen in the channel Monte Carlos without significant fine tuning. We instead opt for a *nearest neighbor* approach.

Since we are only interested in evaluating the PDF for each event channel k at the values for which we have data, and not for arbitrary data values, we abandon the task of estimating the PDF for an arbitrary value x. This reduces the problem from estimating a function to estimating a set of numbers $P^k(\vec{x}) = \{P^k(x_i)\}$ for the kth event channel. For each data point x_i and channel k, we calculate a probability density by counting the number of nearby Monte Carlo samples N_s^k within a range r_s, then dividing by r_s and the total number of Monte Carlo samples N_{tot}^k.

$$P^k(x_i) = \frac{N_s^k(x_i - \frac{r_s}{2} < x < x_i + \frac{r_s}{2})}{N_{tot}^k r_s} \tag{1}$$

We enforce normalization by computing the Riemann sum

$$\begin{array}{c} \Sigma_i^{n_{events}} P^k(x_i)\Delta_i = 1 \\ \Delta_i = \frac{x_{i+1}+x_i}{2} - \frac{x_i+x_{i-1}}{2}. \end{array} \tag{2}$$

The estimate of the PDF will have noise from the finite size of the Monte Carlo sample. In addition, the values $P(x_i)$ will be correlated since the same points are used for neighboring density estimations. To account for these effects, we produce bootstrap realizations of the Monte Carlo and calculate PDFs for each random draw. The ensemble of $\{P_j^k(\vec{x})\}$, where j indexes bootstraps, are random draws from the PDF of $P^k(\vec{x})$ that includes the uncertainty from both finite number and correlation.

2.2 Measuring Signal Rates

We perform a modified maximum likelihood analysis to measure the fraction of events N from each channel k_{b1}, k_{b2}, and k_s. To enforce $\Sigma_i k_i = 1$, we assign the following priors, with $U(a,b)$ meaning uniform probability between a and b.

$$\begin{array}{rcl} k_{b1} & \sim & U(0,1) \\ k_s & \sim & U(0,1-k_{b1}) \\ k_{b2} & = & 1-k_{b1}-k_{b2} \end{array} \tag{3}$$

We apply Bayes' theorem to derive the basic form of the analysis, where N is the number of events in the data set, $\vec{k} = \langle k_{b1}, k_{b2}, k_s \rangle$ is a vector of the fractions described above, $\vec{D} = \{x_i\}$ represents the measured values in the data set, and $\vec{P}(\vec{x}) = \langle P^{k_{b1}}(\vec{x}), P^{k_{b2}}(\vec{x}), P^{k_s}(\vec{x}) \rangle$ are the probabilities measured from each event channel for each event in the data set.

$$P(\vec{k}|\vec{D}) \propto P(\vec{D}|\vec{k})P(\vec{k}) = P(\vec{k}) \int_{\vec{P}} P(\vec{D}|\vec{k}, \vec{P}(\vec{x}))P(\vec{P}(\vec{x}))d\vec{P} \tag{4}$$

The PDF $P(\vec{k})$ is a combination of the priors from Eq 3 and the priors in the problem statement, with $TNorm(x|\mu,\sigma)$ representing a truncated Gaussian distribution.

$$P(\vec{k}) = U(k_{b1}|0,1)U(k_s|0,1-k_{b1})TNorm(k_{b1}N|900,90)TNorm(k_{b2}N|100,100) \tag{5}$$

Similar to a standard maximum likelihood analysis, the probability that a channel k produced an event with value x_i is $P^k(x_i)$. Therefore the likelihood takes the following form, where the dot product $\vec{k} \cdot \vec{P}(x_i)$ is a sum over each channel in the fit.

$$P(\vec{D}|\vec{k}, \vec{P}(\vec{x})) = \Pi_i^N P(x_i|\vec{k}, \vec{P}(x_i)) = \Pi_i^N \vec{k} \cdot \vec{P}(x_i) \tag{6}$$

As seen above, the probability $P(\vec{k}|\vec{D})$ depends on the probabilities measured for each event in each channel, $\vec{P}(\vec{x})$. However, as described in section 2.1, the $\vec{P}(\vec{x})$ have correlated uncer-

tainties. We also do not care what the actual values of $\vec{P}(\vec{x})$ are. We therefore marginalize over $\vec{P}(\vec{x})$ in Eq 4.

2.3 Model Selection

We need to infer the presence or absence of the signal channel. This is a model selection problem between $k_s = 0$ and k_s as a free parameter, corresponding to no signal and signal, respectively. Standard Bayesian practice is to compute the ratio of the Bayesian evidence between the two models. Calculating the Bayesian evidence involves integrating over all parameters, which may be computationally costly and numerically unstable without proper Monte Carlo sampling procedures. For reasons of simplicity and time, we elect to instead compute the ratio of the probabilities of each model at the best fit parameters, *i.e.* the delta log-likelihood . This is justified in the limit that the posterior is uni-modal and peaky around the best fit value, dominating the integral in the Bayesian evidence. We calibrate the delta log-likelihood statistic by measuring its distribution in zero-signal toy Monte Carlo simulations.

2.4 Implementation

We use the Minuit minimizer to find best fit parameters for the model. We minimize the negative log form of equation 4.

$$\begin{aligned} -\log P(\vec{k}|\vec{D}) &= K - \log TNorm(k_{b1}N|900, 90) \\ &\quad - \log TNorm(k_{b2}N|100, 100) \\ &\quad - \log(\Sigma_j^{N_{bootstraps}} \Pi_i^{N_{events}} \vec{k} \cdot \vec{P}_j(x_i)) \end{aligned} \tag{7}$$

In the above equation, K is the normalization constant, which we set to 0. The marginalization over $\vec{P}(\vec{x})$ is evaluated by Monte Carlo integration, where we use the bootstrap samples described in section 2.1 as our sample points. This numeric integration at each parameter value forces us to assume the unusual form of Eq 7 where we explicitly multiply the probabilities of each event (which may lead to errors from finite precision) instead of the usual form $\Sigma_i^N \log P(x_i|\vec{k})$. We would like to stress that the bootstraps **are not** over the data points measuring in each data set. Instead, we bootstrap the event channel Monte Carlo and measure a PDF $P^k(\vec{x})$ for each random draw as described in section 2.1.

3 Toy studies

We study the effectiveness of this discriminating procedure by running over many toy datasets. We randomly select datasets from the MC samples provided: background 1 (bkg1), background 2 (bkg2), and signal (sig). We sample from the background distributions according to the priors described in the problem description: a truncated Gaussian with mean of 900 and width 90 for bkg1 and a truncated Gaussian with mean of 100 and width 100 for bkg2. We generate 1000 datasets with this cocktail, taking care not to repeat events within a given dataset. We also generate 1000 different datasets using the same sampling recipe (but not the exact same events) for the backgrounds, but this time we embed exactly 75 random events from the sig dataset.

We then fit these datasets using the technique described in the previous section. We run over the datasets with different values for the *range* (r_s) in which to count the nearest neighbors: 0.005, 0.010, and 0.020. We try different values for the number of bootstrap samples to generate: 10, 100, and 1000. We also run over the datasets where we do not use the bootstraps, and instead only use the PDF calculated from the original provided MC samples. This is referred to as the 0 bootstrap set of trials.

3.1 Significance of observation of signal

For each dataset, we run two fits: one fit where we allow the procedure to fit for bkg1, bkg2 and sig, and one fit where we set sig=0. We look at the different in the negative log likelihoods (NLL) and use a fairly common definition for the significance (σ) of the observed signal, where the likelihood function is approximated as Gaussian around the minimum.

$$\sigma = \sqrt{2\Delta NLL}$$

We do not interpret this quantity as a rigorous derivation of the width of a Gaussian around the minimum. Instead, we run over the 1000 toy datasets with 0 signal events, and find the value below which 99% of the fits lie. We interpret this as the point at which we could assume a 1% rate for Type I errors. This value is referred to as σ_{99} in later sections.

For each dataset with 75 events, we find σ_{99} for that # of bootstraps and r_s, and count how many of the datasets lie above this value. We interpret this fraction as the *discriminating power* of the procedure.

3.2 Failed fits

We found that for the toy fits where there were 0 signal events and 10,100, or 1000 bootstraps were generated, 20-27% of the fits failed when the procedure had the freedom to add signal events. However, the fit where no signal events were assumed would succeed for almost all (less than 2 out of 1000 failed) of the datasets. For the purposes of our study, we assigned these fits to have 0 significance and included them in our determination of σ_{99}.

For the 0 bootstrap samples, about 50% of the fits failed in this fashion.

The "failure" of the fits, were for one of two reasons: either Minuit was not able to calculate a good error matrix, even though the values to which it converged looked reasonable, or there seemed to be some precision issue such that the NLL was *very* small, but slightly negative and so the square root would return `nan`. It may be that there was some normalization issue in the code that we did not track down, but with limited time we chose to simply include these fits as "good" with 0 significance.

3.3 Results from toy studies

The distributions for the σ quantity for both the 0 signal events datasets and the 75 signal events datasets can be seen in Figs. 1-3 for the tested values of # of bootstraps and r_s. Dashed lines in the figures represent the σ_{99} value.

Table 2 summarizes the results of these studies showing σ_{99} and the power for each of these studies. These values are shown in graphical form in Fig. 4 and Fig. 5.

It is questionable whether 1000 toy studies is enough to discern any significant differences between the different settings for the fit (# bootstraps, r_s). We note that for the # bootstrap=1000 studies, we appear to lose some discriminating power as r_s is increased. This is perhaps to be expected as increasing r_s means one loses sensitivity to rapid changes in the distribution of MC events across the x-axis.

All of the studies yield fairly consistent results, although we stress we do not actually know how significant these differences are. However, we need to pick a particular setting to use with the datasets and so we settle on the one that gives us the greatest discriminating power according to these studies: # bootstraps=1000 and $r_s = 0.005$, which yields a discriminating power of 95.3% with a Type I error of 1%.

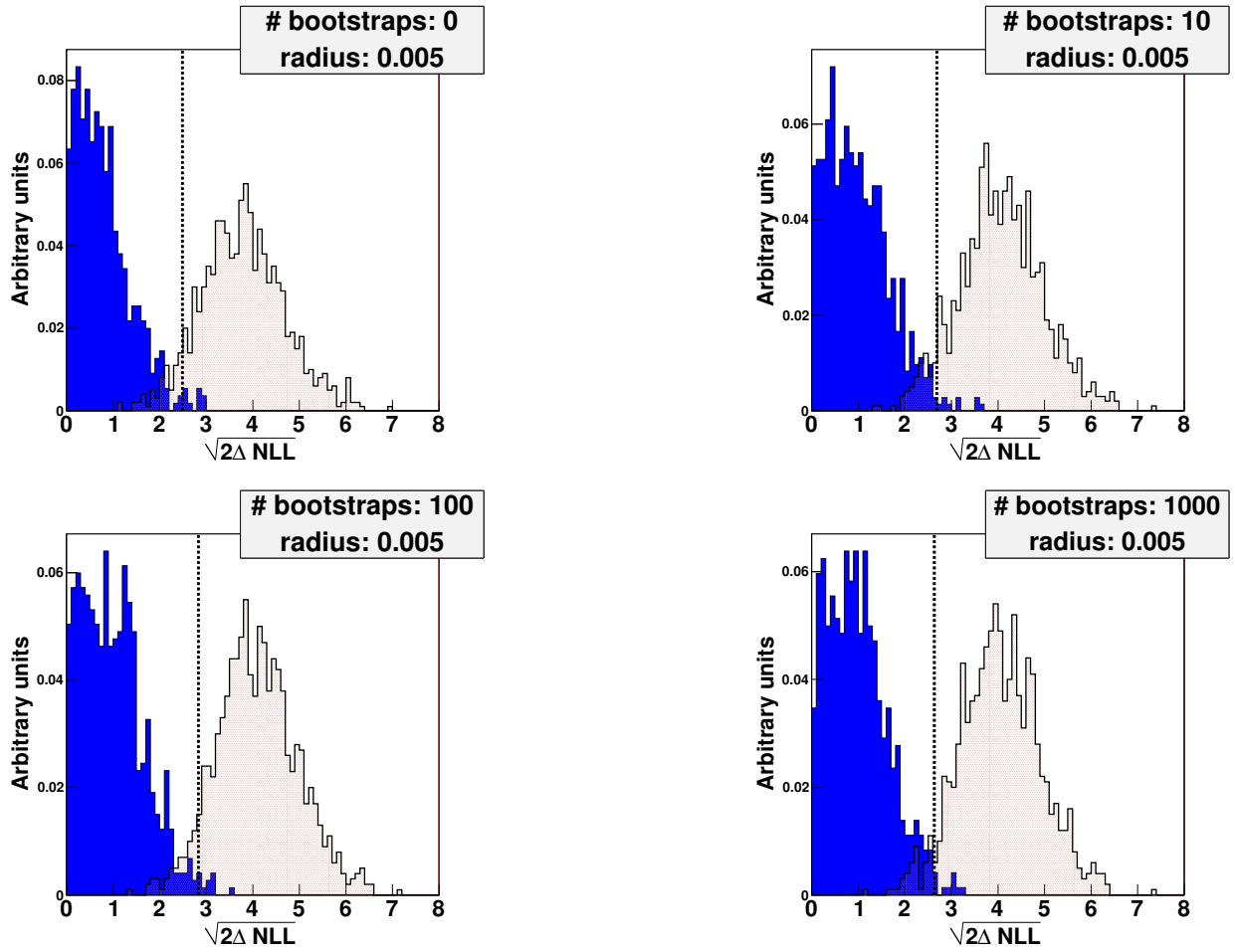

Figure 1: Distribution of significance for 1000 toy studies with 0 signal events embedded (blue) or 75 signal events embedded (shaded tan). Studies were run with 0, 10, 100, and 1000 bootstrap samples, as indicated in each figure. These studies use 0.005 as the range, r_s, in which to count the nearest neighbors, as described in the text. There are some datasets in the 0 signal events sample which are not plotted here due to a pathology in the fitting routine. However, they are counted as a 0 significance dataset. The dashed line indicates the point below which 99% of the 0 signal event datasets lie. Note that the areas are normalized to be the same for each distribution, including underflows and overflows.

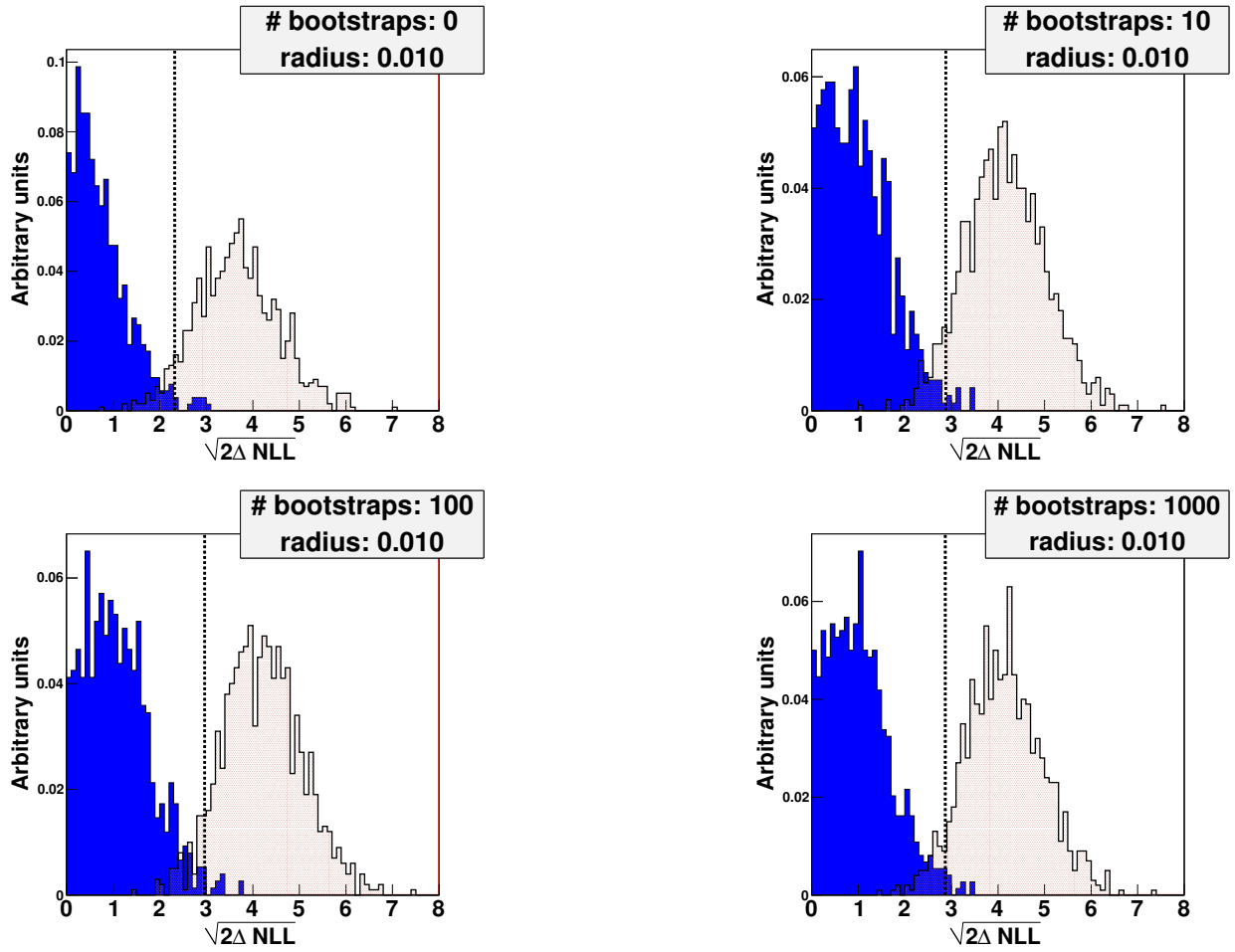

Figure 2: Distribution of significance for 1000 toy studies with 0 signal events embedded (blue) or 75 signal events embedded (shaded tan). Studies were run with 0, 10, 100, and 1000 bootstrap samples, as indicated in each figure. These studies use 0.010 as the range, r_s, in which to count the nearest neighbors, as described in the text. There are some datasets in the 0 signal events sample which are not plotted here due to a pathology in the fitting routine. However, they are counted as a 0 significance dataset. The dashed line indicates the point below which 99% of the 0 signal event datasets lie. Note that the areas are normalized to be the same for each distribution, including underflows and overflows.

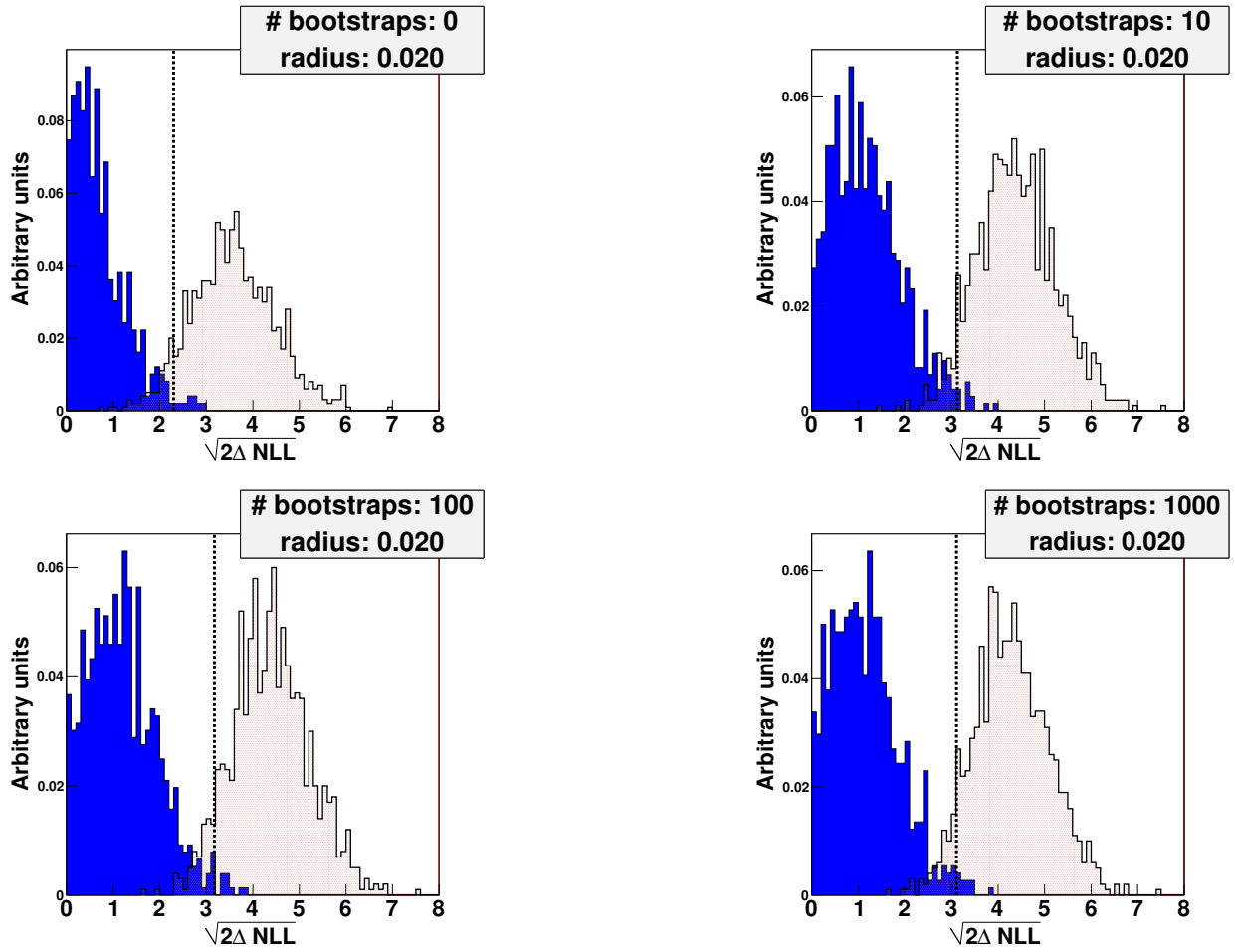

Figure 3: Distribution of significance for 1000 toy studies with 0 signal events embedded (blue) or 75 signal events embedded (shaded tan). Studies were run with 0, 10, 100, and 1000 bootstrap samples, as indicated in each figure. These studies use 0.020 as the range, r_s, in which to count the nearest neighbors, as described in the text. There are some datasets in the 0 signal events sample which are not plotted here due to a pathology in the fitting routine. However, they are counted as a 0 significance dataset. The dashed line indicates the point below which 99% of the 0 signal event datasets lie. Note that the areas are normalized to be the same for each distribution, including underflows and overflows.

Table 1: Summary of toy studies. For each of the studies, 1000 datasets with 0 signal events and 1000 datasets with 75 signal events were fit to two hypothesis: that the dataset consisted of some cocktail of the two background and the signal or some cocktail of only the two backgrounds. Shown is the σ_{99} quantity and the power for the procedure given some # of bootstraps and range, r_s, used in the fit. See text for more description of this procedure.

# bootstraps	r_s	σ_{99}	Power (fraction)
0	0.005	2.48	0.933
0	0.010	2.32	0.942
0	0.020	2.29	0.926
10	0.005	2.68	0.939
10	0.010	2.87	0.935
10	0.020	3.12	0.946
100	0.005	2.83	0.936
100	0.010	2.96	0.938
100	0.020	3.17	0.933
1000	0.005	2.63	0.953
1000	0.010	2.86	0.941
1000	0.020	3.10	0.936

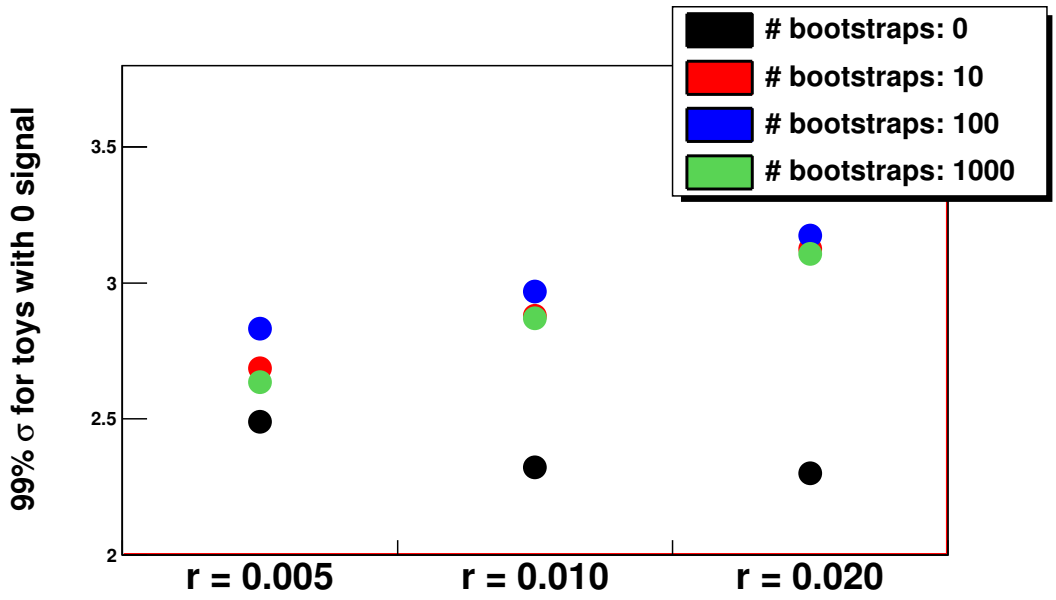

Figure 4: Summary of toy studies with 0 signal events embedded. For a given toy study where n bootstrap samples were run using r as the r_s in which to find the number of nearest neighbors, we plot the value of σ ($\sigma = \sqrt{2\Delta NLL}$) below which 99% of those datasets lie. This is interpreted as the value which would yield a 1% Type I error rate.

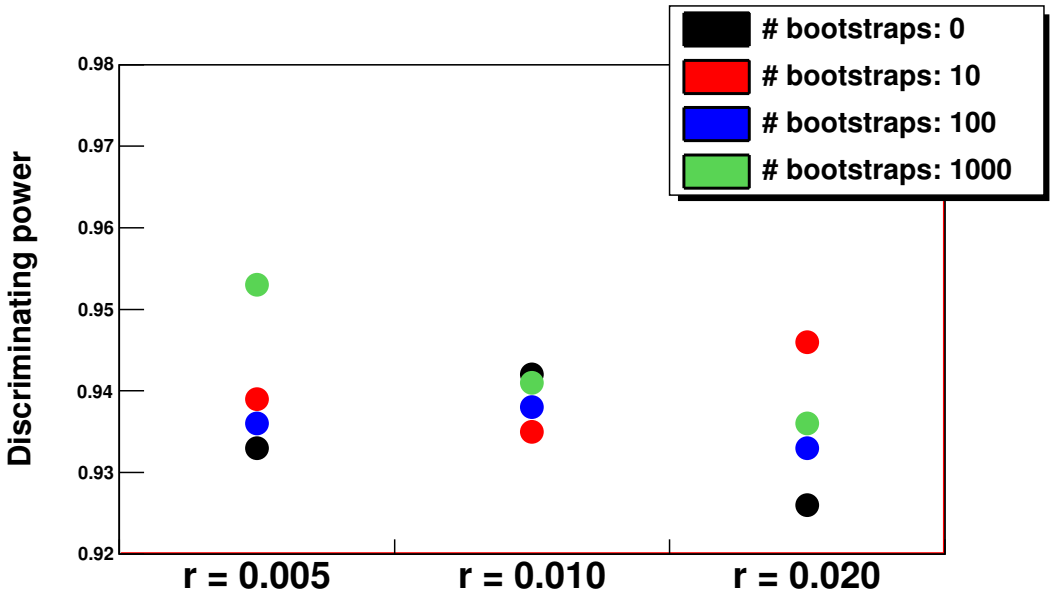

Figure 5: Summary of toy studies with 75 signal events embedded. For a given toy study where n bootstrap samples were run using r as the r_s in which to find the number of nearest neighbors, we plot the fraction of datasets which lie above the value of σ ($\sigma = \sqrt{2\Delta NLL}$) which gives a 1% Type I error rate for that combination of n and r.

Table 2: Summary of fits to the data samples. Shown is the σ_{99} quantity and the power for the procedure given some # of bootstraps and range, r_s, used in the fit. See text for more description of this procedure.

# bootstraps	r_s	σ_{99}	Power (fraction)
100	0.005	2.83	0.101
100	0.010	2.96	0.101
1000	0.005	2.63	0.108
1000	0.010	2.86	0.103

4 Results from the "data"

While we have settled on a particular settings to use for the 20,000 "real" datasets, we run studies with # bootstraps to be 100 and 1000 and r_s to be 0.005 and 0.010. The distributions for σ is shown in Fig. 6 and Fig. 7, along with the distributions for the toy studies with 0 signal events. Note that the areas are normalized to be the same for each distribution, including underflows and overflows. The summary of these results is shown in Table 2 and in graphical form in Fig. 8 and Fig. 9.

We find that about 10% of the datasets show a significant signal for all settings of the fit. We also note that 25-27% of the fits "fail" as described in the earlier section. These fits are assigned 0 for the significance and interpreted as not containing signal events.

The final results for the 20,000 datafiles will be for # bootstraps = 1000 and $r_s = 0.005$. The determination of a significant signal or not will be for $\sigma > 2.63$, where σ is described earlier in the text. The filename containing the final results is `final_results.txt` and the format is a text file with three columns: dataset number, "yes" or "no" for the question of signal, and the significance (σ) returned by our procedure for that dataset. A snippet of the results file follows.

```
...
329  no 0.000
330 yes 3.806
331  no 1.868
...
```

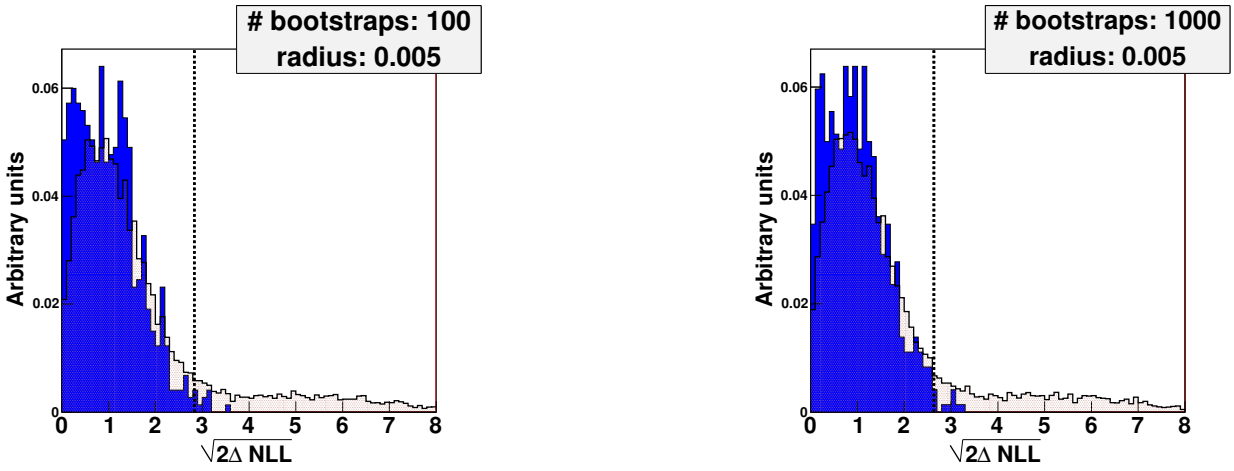

Figure 6: Distribution of significance for the provided datasets. Shown are the 1000 toy studies with 0 signal events embedded (blue) and the 20000 provided datasets (shaded tan). Studies were run with 100 and 1000 bootstrap samples, as indicated in each figure. These studies use 0.005 as the range, r_s, in which to count the nearest neighbors, as described in the text. There are some datasets in the 0 signal events sample which are not plotted here due to a pathology in the fitting routine. However, they are counted as a 0 significance dataset. The dashed line indicates the point below which 99% of the 0 signal event datasets lie. Note that the areas are normalized to be the same for each distribution, including underflows and overflows.

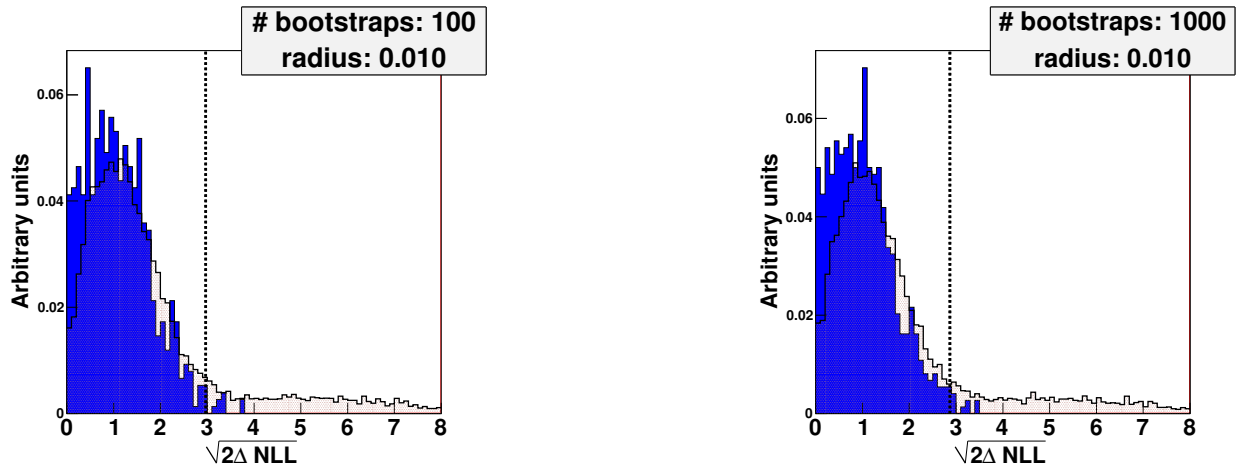

Figure 7: Distribution of significance for the provided datasets. Shown are the 1000 toy studies with 0 signal events embedded (blue) and the 20000 provided datasets (shaded tan). Studies were run with 100 and 1000 bootstrap samples, as indicated in each figure. These studies use 0.005 as the range, r_s, in which to count the nearest neighbors, as described in the text. There are some datasets in the 0 signal events sample which are not plotted here due to a pathology in the fitting routine. However, they are counted as a 0 significance dataset. The dashed line indicates the point below which 99% of the 0 signal event datasets lie. Note that the areas are normalized to be the same for each distribution, including underflows and overflows.

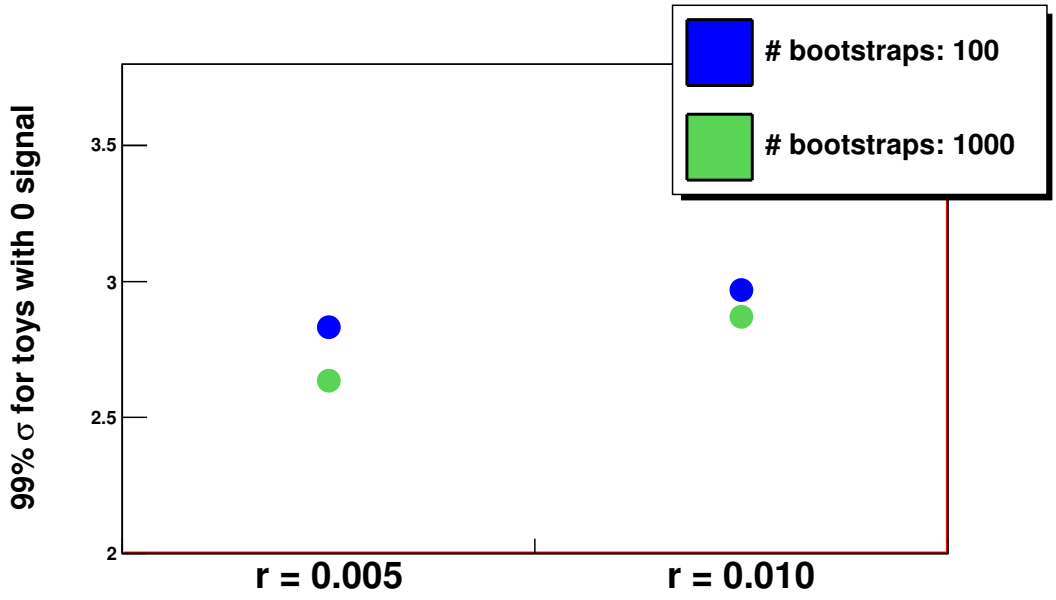

Figure 8: Distribution of significance for 1000 toy studies with 0 signal events embedded (blue)

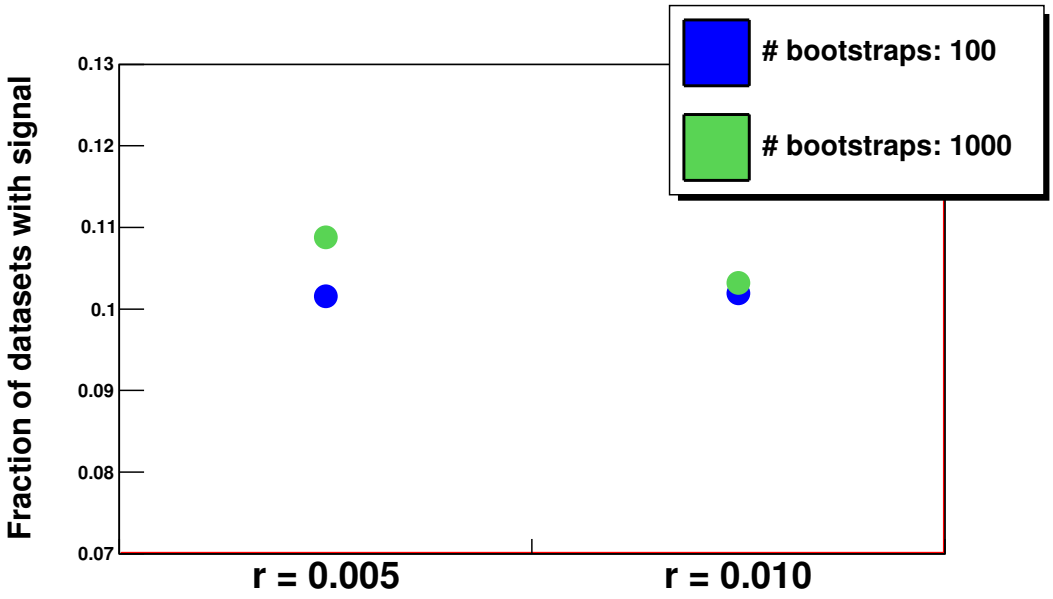

Figure 9: Distribution of significance for 1000 toy studies with 0 signal events embedded (blue)

5 Summary

The nearest neighbor "fit" yields a 95% discriminating power at a 1% Type I error rate for samples with 75 signal events. The results from applying this procedure to the 20,000 datasets can be found in the attached text files.

There are still some outstanding questions that were not addressed by this study, primarily because of time constraints on the authors.

- The issue of the "failed" fits when there are 0 signal events, but the fit has the freedom to include signal. This could probably be addressed with some fine tuning of the fitting code.
- Length of time to run a fit. When we use 1000 bootstraps, it takes about 2.0-2.5 minutes to run both fits (2 bkg and 1 sig or just 2 bkg) on a dataset with approximately 1000 events. The time sink is calculating the nearest neighbors for all 1000 bootstrap samples. We run our jobs on the SLAC batch queue, so we are able to complete a MC study in a reasonable amount of time. However, if one wanted to optimize this procedure, there are probably code improvement that could be made or the nearest neighbor calculation could be moved to the GPU using CUDA or a similar programming language.
- The significance of the differences between the different fit options used in the toy studies was not addressed by this analyses. One would want to run more MC to test this (10,000 datasets? 100,000 datasets?)

In the end, this has been a very interesting exercise which forced us to examine some of these statistical issues at a closer level than we have previously done. We thank the organizers for taking the time to put together this challenge.

6 About the authors

Matt Bellis is a post-doc research associate at Stanford University working with Pat Burchat on analysis of BaBar data. He is currently wrapping up a search for baryon- and lepton-number violating decays of B mesons.

Douglas Applegate is a graduate student at Stanford University. He is a member of the X-ray Observational Cosmology group at the Kavli Institute for Particle Physics and Cosmology and is advised by Steve Allen. Douglas measures the mass distributions of massive, X-ray selected galaxy clusters via the gravitational weak lensing effect. Measurements of the mass distributions may be used to calibrate mass proxies in cosmology analyzes, constrain non-thermal pressure support in the inter-cluster medium, and investigate properties of Dark Matter in galaxy clusters.

Previously, Douglas was an undergraduate and Matt was a post-doc in Carnegie Mellon's Medium Energy Group, working with Curtis Meyer and Mike Williams.

We thank Mark Allen at Stanford, for many useful discussions.